Map management for robust long-term visual localization of an autonomous shuttle in changing conditions

Youssef Bouaziz, Eric Royer, Guillaume Bresson, Michel Dhome

Noname manuscript No.
(will be inserted by the editor)

Map management for robust long-term visual localization of an autonomous shuttle in changing conditions

Youssef Bouaziz · Eric Royer · Guillaume Bresson · Michel Dhome

Received: date / Accepted: date

Abstract Appearance changes are one of the most challenging problems for the visual localization of an autonomous vehicle in outdoor environments. Data association between the current image and the landmarks in the map can be difficult if the map was built with different environmental conditions. This paper presents a solution to build and use multi-session maps incorporating sequences recorded in different conditions (day, night, fog, snow, rain, change of season, etc.). During visual localization, we exploit a ranking function to extract the most relevant information from the map. This ranking function is designed to take into account the pose of the vehicle and the current environmental condition. In the mapping phase, covering all conditions by constantly adding data to the map leads to a continuous growth in the map size which in turn deteriorates the localization speed and performance. Our map management strategy is an incremental approach that aims to limit the size of the map while keeping it as diverse as possible. Our experiments were performed on real data collected with our autonomous shuttle as well as on a widely used public dataset. The results demonstrate that our approach has significantly improved localization performance in different challenging conditions.

Y. Bouaziz
Institut Pascal, CNRS, SIGMA Clermont, Clermont-Ferrand, France
Institut VEDECOM, Versailles, France
E-mail: youssef.bouaziz@etu.uca.fr

E. Royer
Institut Pascal, CNRS, SIGMA Clermont, Clermont-Ferrand, France
E-mail: eric.royer@uca.fr

G. Bresson
Institut VEDECOM, Versailles, France
E-mail: guillaume.bresson@vedecom.fr

M. Dhome
Institut Pascal, CNRS, SIGMA Clermont, Clermont-Ferrand, France
E-mail: michel.dhome@uca.fr

Keywords Visual-Based Navigation · Computer Vision for Transportation · SLAM

1 Introduction

Autonomous navigation in dynamic environments is becoming a central issue in robotic research field. Accurate localization in such environment is a must for mobile robots to ensure reliable and safe navigation. Simultaneous localization and mapping (SLAM) offers a decent solution for mobile robots to perform localization and mapping without any prior knowledge of the environment. SLAM is a process in which an autonomous robot models its own environment using different kinds of sensors, while simultaneously, predicting its own position in this map.

Recently, cameras are becoming one of the most commonly used types of sensors in SLAM thanks to their cheap setup requirements and their efficiency. Visual-SLAM (SLAM using cameras as sensors) has widely drawn attention of researchers and it is seeing a growing number of real-time applications in different disciplines. Appearance change remains a major challenge for visual-SLAM, re-localization in previously mapped areas can be a hard task since the appearance of the environment is changing ceaselessly and such phenomenon can result in inaccurate or erroneous localization. In some applications like self-driving cars, the pose estimation part of the SLAM process is very delicate and must be taken very carefully because even a small error in estimating the pose of the vehicle can result in a dangerous accident. To ensure a secure and safe life-long navigation in dynamic environments, autonomous vehicles must cope with such challenge.

In this work, we are developing a SLAM system for autonomous shuttles. In such context, shuttles are frequently revisiting the same place in different times and different environmental conditions. In some of our previous works [29], we employed a driverless shuttle for three months on an industrial site, totaling nearly 1500 km of autonomous travel. During this experience, we identified some long-term difficulties for autonomous navigation. Operating in dynamic environments for long period can considerably impact localization performance over time. Localization in such scenario using a primitive mapping system where no map management is involved will result in a continuous growth of the map. Therefore, a large memory space is needed to store such map resulting in an exponential increase of the time required to retrieve relevant information for localization.

This paper is an extension of our previous work published in a conference paper [2] in which we presented a keyframe retrieval technique able to retrieve relevant data from a multi-session map that incorporates multiple environmental conditions. In this present paper, we introduce two new contributions compared to our previously published work:

– an extended evaluation of the keyframe retrieval algorithm on a much larger dataset.

– a map management algorithm which builds a multi-session map targeted at optimal localization performance with a bounded memory constraint.

We will recall the keyframe retrieval approach in this paper for a better understanding of the new evaluations. In this keyframe retrieval approach, we proposed a localization approach able to take advantage of a visual landmark map composed of N sequences gathered at different times. Generally, basic SLAM algorithms retrieve the keyframes that are used for localization based on their geometric distance to the vehicle pose. However, this technique has shown a weakness in long-term localization because it doesn't take into account environmental changes. This incents us to develop a new strategy for keyframe retrieval in order to adapt our SLAM algorithm [15], [29] for long-term operation. During the localization process, we aim to maximise the number of matched points (i.e. ameliorate the localization performance) by retrieving relevant experiences from a map that integrates numerous environmental conditions. Our proposed keyframes retrieval approach takes advantage of statistics gathered in the first few meters of a traversal in a given location to compute a probabilistic ranking function. This ranking function is used in the rest of the traversal to retrieve from the map the most relevant keyframes, taking into account the current environmental conditions. In order to ensure our ranking function consistency, we keep updating it regularly throughout the trajectory.

The second contribution of this paper consists of a map management approach. In this approach, we aim to extend the keyframe retrieval technique by adding a complementary algorithm that is designed to reduce the size of the map. This algorithm is based on some fundamentals proposed in the keyframe retrieval approach. The aim of the map management approach is to maintain a reliable map with a fixed size throughout the frequent runs. In this part, we exploit the scores of resemblance between the traversals in the map. These scores were computed in the keyframe retrieval part and will be used to determine which traversal has the highest similarity to the others to remove it eventually.

In Figure 1, we present a diagram explaining the operating mechanism of both processes and the link between them. This diagram is constituted from two main parts, the keyframes retrieval part (recalled in Section 3), and the map management part explained in Section 4. Our system uses a multi-session map that incorporates multiple traversals recorded in different environmental conditions. Each time a new image arrives, the keyframes retrieval algorithm has to extract from the map relevant data to the current environmental condition. These retrieved data are then used to compute the vehicle pose. After the end of the localization session, our map management algorithm will be invoked offline to update the map by adding new data or removing superfluous information.

We evaluated our approaches on the Oxford RobotCar dataset [19] and a new dataset recorded on our vehicle that we make available to the community.

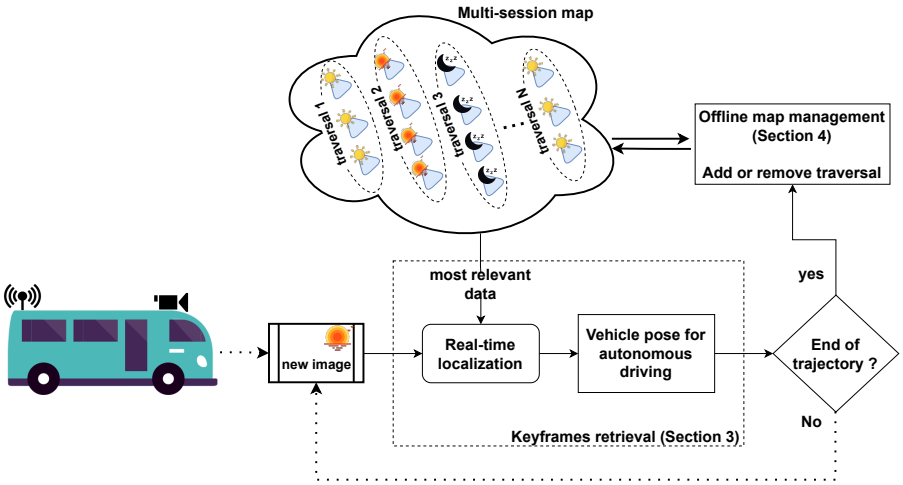


Fig. 1: A diagram representing the operating mechanism of the approach proposed in this paper.

Our dataset is called IPLT[1](Institut Pascal Long-Term) dataset and it contains, at the moment, 109 sequences recorded over a 16 months period in which the vehicle has followed the same path around a parking lot with slight lateral and angular deviations. This dataset contains various environmental conditions due to changes in luminance, weather, seasons and in parked vehicles and each sequence is around 200 m length (see Figure 2).

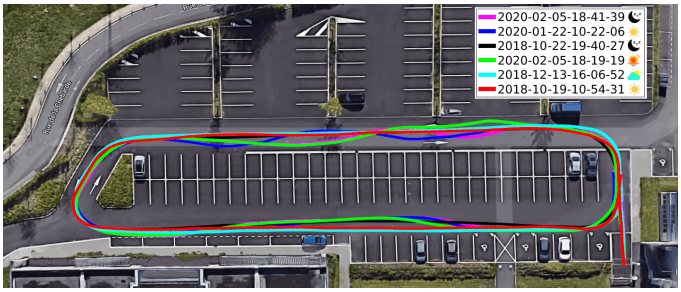


Fig. 2: Example of 6 sequences recorded in a parking lot.

Our experiments demonstrated that our approach has significantly increased localization performance and allowed us to successfully localize in challenging conditions.

[1] To download our dataset please visit `http://iplt.ip.uca.fr/datasets/` and enter the following username/password for a read-only access to our ftp server: ipltuser/iplt_ro

2 Related work

Substantial efforts were made in visual recognition of places in static scenes or with few moving objects, but it is only recently that attempts have been made to extend localization performance in changing environments (for example, day-to-night or season-to-season correspondence).

Traditional feature-based comparison techniques are not suitable for long-term operations due to their weakness to changing conditions. An image based approach was proposed by Murillo and Kosecka [24] to improve localization in dynamic environments by representing a place by a global descriptor calculated with the entire image. In this approach, recognizing a place requires a deep search in the database to find the corresponding image, which is costly in large-scale environments. Milford and Wyeth [21] proposed to enhance the performance of global image descriptors for places recognition by matching sequences («SeqSLAM») of images instead of unique images and they achieved impressive results on various seasonal datasets. Although «SeqSLAM» has shown excellent performance in some situations, it remains sensitive to viewpoint changes. Pepperell *et al.* [27] extended the classic SeqSLAM structure that uses linear image databases. They integrated an oriented graph structure to represent the roads. They also used panoramic images to minimize the variance of viewpoints. Despite all these improvements, the global image methods still remain sensitive to significant variations in viewpoint. Moreover, these methods cannot provide a 6DoF (Degrees of Freedom) estimation of the vehicle pose. Recently, new deep learning approaches such as [25], [30], [28] were proposed to improve localization robustness in challenging conditions. However, high-end GPUs are required to achieve real-time localization and most of these approaches do not allow to estimate the 6DoF pose of the vehicle. Some other works like LIFT [34], NID-SLAM [26], LUIFT [10] and SOS-Net [32] highlighted descriptors effect in localization with brightness changes and proposed new descriptors which are more robust in such conditions than SIFT [17] and SURF [1]. These works can be combined with other map management approaches to improve the robustness of localization against changes in environmental conditions.

Muhlfellner *et al.* [22] proposed a landmark selection approach, in which, a score is assigned to each landmark based on the number of times it was observed. Those scores are used in the localization phase to choose the most relevant landmarks. Bürki *et al.* [5] exploited statistics calculated in past traversals to create a ranking function that helps to select only the relevant landmarks for a given environment condition. However, this approach reaches its limits on maps containing different conditions at the same time. In more recent works of Burki *et al.* [4], landmarks which are used in localization are updated if the localization performance exceeds a predefined threshold, otherwise, new points are added to the map. Both in [4] and in the proposed approach of Dymczyk *et al.* [11], the size of the map is controlled by performing an offline map maintenance process which aims to produce a reliable map with a fixed size. More recently, Bürki *et al.* [3] exploited some popular infor-

mation retrieval methods from other fields like document retrieval approaches using text queries to search for relevant landmarks to the current environment conditions. MacTavish *et al.* [18] have also addressed problem of long-term localization in a similar way, they used a collaborative filtering based approach that identifies experiences based on the landmark matching history.

In order to build multi-session map, Churchill and Newman [7] proposed an approach in which a place can have different appearances. They developed a mapping system based on a "plastic" map (a compromise between adapting to new models and preserving old models). This structure allows to memorize different experiences for each place rather than trying to match different appearances between seasons and/or brightness changes. They also proposed a similar approach [8] in which they added new experiences whenever a localization failure occurred in some mapped area. However, in both [7], [8], the size of the map varies according to the variations of the scene and the query image must be matched to all experiences to find the best match. More recently, the works of Linegar *et al.* [16] were devoted to reduce computational costs of [8]. To do so, they exploited past experiences of successful localizations to recall similar experiences under the current environmental condition. This approach is computationally inexpensive because it does not directly take into account the appearance information. However, it cannot effectively select future experiences unless there are enough past experiences stored in the history.

We approach the problem of long-term localization in another way. We exploit information collected in the beginning of the traversal to compute a ranking function that retrieves relevant landmarks taking into account both environmental conditions and geometric distance to the closest keyframes in the map. This ranking function is also used in an offline map management step in which we first calculate a similarity matrix that stores the resemblance scores between all the traversals in the map. Then, we analyze all this stored scores in order to remove similar traversals so we can produce a reliable map with a bounded size.

3 Keyframes retrieval

It is difficult to estimate the probability of matching features between two images with different viewpoints and which were not taken under the same environmental condition. We developed a re-localization process able to perform on a global map consisting of N traversals (experiences) recorded in different conditions. For example, the map presented in the Section 6.1.1 was built with the 10 traversals shown in Figure 7. The map was generated with an experience-based mapping framework [15], [29] based on keyframes and local features and it is very similar to some well-known frameworks such as ORB-SLAM [23] and Maplab [31]. The mapping phase of our system starts with interest points detection and matching. We extract a keyframe from the video flow every ~ 1 meter, and for each keyframe, we compute its 6DoF pose and the set of its related 3D points through triangulation. Therefore, a keyframe

is added to the map by storing its extracted features and its corresponding 6DoF pose (similarly, a keyframe is retrieved from map by retrieving its corresponding features and 6DoF pose). All the stored features will be linked with their corresponding 3D points in the map. To make sure that we build an experience-based map, all keyframes belonging to a same run are grouped in a same structure called traversal. This will make easier the access to the features recorded in a particular run (benefits of experience-based mapping). Moreover, our mapping system also involves an online loop-closure operation which detects and closes loops in the map. In the localization phase, our system consists, in a first step, in predicting the current pose using the last computed pose and the odometry input. Afterwards, the most relevant keyframe according to the predicted pose will be retrieved from the map to perform a 2D/3D matching with the detected interest points from the current input image. Finally, we use RANSAC and PnP (Perspective-n-Point) to compute the current pose of the vehicle which will be optimized in the last step.

In order to retrieve only relevant information for localization, we designed a ranking function that is partially built with offline data. The role of this function is to retrieve from the map the most relevant keyframes taking into account two main factors:

- The geometric distance between the vehicle pose estimation and the pose associated to the keyframe K_j which is used for localization
- The environmental conditions of the keyframe K_j (lighting, weather, season...).

The ranking function is computed during the first few meters of the trajectory. After that, it will be used to estimate the probability that a point pt extracted from the current image I_i finds a match in the keyframe K_j retrieved from the map. The ranking function is described by the Equation (1):

$$P(pt \in I_i, K_j) = f_{dist}(I_i, K_j) \cdot f_c(I_i, K_j) \tag{1}$$

$f_{dist}(I_i, K_j)$ is a function defining a score for matching the current image I_i with the keyframe K_j retrieved from the map by supposing that both of them were taken in similar conditions (weather, lighting...). This implies that computing this score depends only on the geometric distance between the two images. On the other hand, $f_c(I_i, K_j)$ supposes that the poses of the two images are identical; hence it takes into account only the appearance changes to assign the score of matching between the two images I_i and K_j.

f_{dist} was computed offline with the use of some data collected specifically for this purpose. The calculation of $P(pt \in I_i, K_j)$ and the calculation/update of f_c are performed online during the re-localization process.

In Figure 3, we present a diagram that explains the localization process using the proposed ranking function. The ranking function ($P(pt \in I_i, K_j)$) takes as input the current image (and its corresponding predicted pose according to the odometry data) to retrieve from the multi-session map the keyframe (and its associated landmarks) that have the highest similarity score. A new pose

will be computed by matching the current image and the retrieved keyframe. To ensure the consistency of f_c with the environmental condition changes, it will be updated along the traversal. For this task, we retrieve from the map another keyframe (each time from a different traversal in a circular way) and use it to update f_c (will be described in more details in Section 3.2.3). The following sections present details on both offline and online computations.

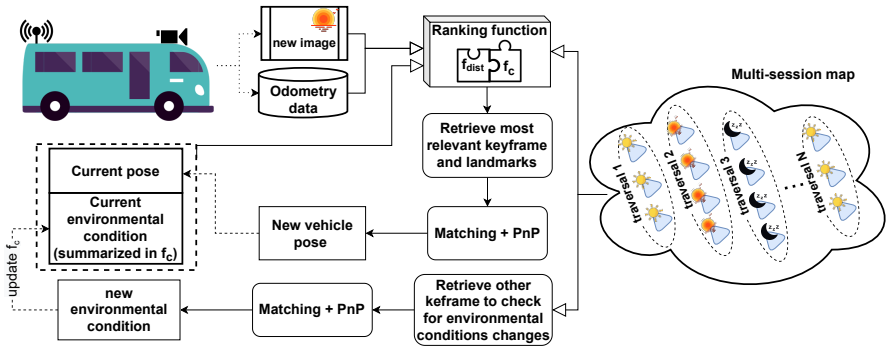


Fig. 3: A diagram representing the operating mechanism of the localization part.

3.1 Offline computation of f_{dist}

We have recorded some sequences specifically for this step. These sequences were recorded successively in a short time period to avoid variation in lighting because $f_{dist}(I_i, K_j)$ assume that I_i and K_j are taken under the same environmental conditions. The sequences were also taken at the same place and the vehicle has followed a slightly different path with some lateral and angular deviation between each pair of sequences. In Figure 4, we present some examples of sequences used for the calculation of f_{dist}.

Fig. 4: Example of 4 sequences used for the calculation of f_{dist}. These sequences were recorded the same day at 15:16, 15:22, 15:23 and 15:24.

We have used 7 sequences for this calculation. For each pair of sequences <SeqA, SeqB>, we use SeqA to generate a map using our SLAM algorithm, then we use SeqB to perform re-localization on the produced map. While performing re-localization with SeqB, for each current pose p_i, we pick the 20 closest keyframes from the map built with SeqA: $K^1, \ldots, K^{20}$ and we match each of them ($K^l, l \in [1, 20]$) to our current image I_i and calculate its corresponding pose p^l in order to calculate the inlier rate: $inliers/(inliers + outliers)$. Then, we compute the longitudinal, lateral and angular gap between p^l and the current pose p_i. Therefore, for each pose p^l, we record a quadruplet consisting of the percentage of inliers, the longitudinal, lateral and angular distance.

f_{dist} is defined as the function that computes the percentage of inliers giving the longitudinal, lateral and angular distance between the poses. We have chosen to define it as a linear combination of Gaussians as in Equation (2):

$$\begin{aligned} f_{dist}(I,K) &= \sum_{h=1}^{n_g} a_h G_h, \text{ with:} \\ G_h &= e^{-\frac{d_x(I,K)^2}{b_h^2} - \frac{d_y(I,K)^2}{c_h^2} - \frac{d_r(I,K)^2}{d_h^2}} \end{aligned} \tag{2}$$

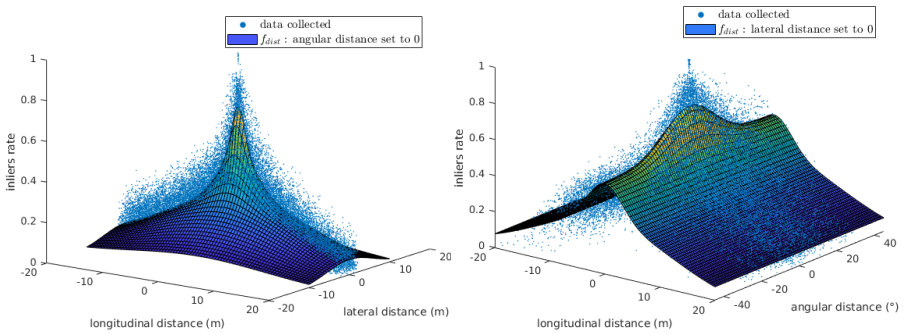
$d_x(I,K)$, $d_y(I,K)$, $d_r(I,K)$ are respectively the longitudinal, lateral and angular (yaw angle) distances between the pose of the image I and the pose of the keyframe K. The parameters a_h, b_h, c_h and d_h are computed with a non linear Least-Squares minimization to fit f_{dist} to the data points. n_g is the number of Gaussians in the model and it was chosen to have the lowest residual error. In Table 1 we show the mean residual error with respect to n_g.

Table 1: Mean of residual errors from fitting f_{dist}.

n	1	2	3	4
Mean	0.090	0.075	0.074	0.074

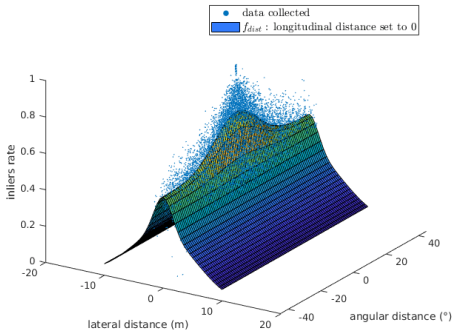
According to the table, we retained n_g=3 as the number of Gaussians in our model. Figure 5 presents the result of fitting the function f_{dist} to the collected data with $n_g = 3$.

The shape of the data collected is dependant on the key-points detector and the features descriptor. In our experiments, we use Harris corner detector [13] for extracting key-points which are matched with ZNCC — Zero-mean Normalized Cross-Correlation — computed on 11×11 pixel windows around each key-point. However, our method can still be applied in the same way using other descriptors.



(a) f_{dist} with $d_r = 0$

(b) f_{dist} with $d_y = 0$



(c) f_{dist} with $d_x = 0$

Fig. 5: Each sub-figure presents the surface of the 4D function f_{dist}. In (a), the inliers rate surface is plotted according to longitudinal and lateral distance while the angular distance d_r is set to 0. The same for (b) and (c), d_y and d_x are set to 0 respectively.

3.2 Online computations

3.2.1 Calculating f_c

We compute the function f_c during the re-localization process. As explained previously, $f_c(I_i, K_j)$ does not depend on the acquisition location of the images I_i and K_j, but only takes into account the condition of the environment to estimate the score of matching between I_i and K_j. Accordingly, we deduce that f_c is traversal dependent, this means that the value of f_c between I_i and any image belonging to the same traversal as K_j will have the same value as $f_c(I_i, K_j)$ (in the normal case where there is no sudden change in the weather or in the lighting condition). Therefore, it is more suitable to design the function f_c as a 2D matrix F_c which have the shape of $[N \times N]$ (with N is the number of traversals existing in the map). The matrix F_c can be formalized as follows: $f_c(I_i, K_j) = F_c(trav(I_i), trav(K_j))$, where the function $trav(X)$ refers to the index of the traversal where the image X belongs (since

I_i is the current image, $trav(I_i)$ refers to the current traversal). The value of $F_c(trav(I_i), trav(K_j))$ corresponds to the score of similarity (which depends on the environmental conditions) between the traversal which contains I_i and the traversal which contains K_j. Thus, Equation (1) becomes:

$$P(pt \in I_i, K_j) = f_{dist}(I_i, K_j) \cdot F_c(trav(I_i), trav(K_j)) \tag{3}$$

In order to define and calculate the matrix F_c, we use our global map (which is composed of N sequences having different environmental conditions) as a reference map to perform a re-localization. During this re-localization phase, we devote the beginning of the trajectory to the calculation of the function. According to Equation (3), we have:

$$F_c(trav(I_i), trav(K_j)) = \frac{P(pt \in I_i, K_j)}{f_{dist}(I_i, K_j)} \tag{4}$$

Our idea consists in calculating the matrix F_c at the start of the trajectory according to Equation (4). Since the ranking function is not yet defined, we measured the inlier rate by matching the image I_i to the keyframe K_j and used it to replace $P(pt \in I_i, K_j)$ in the Equation (4). In the first few meters of the re-localization, we follow the steps described by Algorithm 1 to initialize the values of the matrix F_c.

Algorithm 1: Calculation of the F_c matrix in the first few meters

$c \leftarrow N + 1$ (the index of the current traversal)
Initialize the buffers: $\mathcal{P}[l] \leftarrow \varnothing, \forall l \in [1, N]$
Compute the initial pose p_0 by localizing the image I_0 within the map using bag-of-words.
for each current image I_i in the first few meters **do**
    Predict the pose $\hat{p}_i$ of I_i using the last computed pose p_{i-1} and the wheel odometry data.
    for each traversal l in the map **do**
        Pick from the traversal l the 3 closest keyframes to $\hat{p}_i$: $K_j^l, j \in [1, 3]$
        for each picked keyframe K_j^l **do**
            Match I_i to K_j^l and compute the inlier rate $P(pt \in I_i, K_j^l)$
            Calculate x such as: $x \leftarrow \frac{P(pt \in I_i, K_j^l)}{f_{dist}(I_i, K_j^l)}$
            $\mathcal{P}[l] \leftarrow \mathcal{P}[l] \cup \{x\}$
        end for
    end for
    Calculate pose p_i by matching I_i and the keyframe with the highest number of inliers among the keyframes $\{K_j^l\}, j \in [1, 3], l \in [1, N]$
end for
for each traversal l in the map **do**
    $F_c(c, l) \leftarrow mean(\mathcal{P}[l])$
end for

N is the number of traversals existing in the map.

3.2.2 Calculating the ranking function $P(pt \in I_i, K_j)$

The classic method for keyframes retrieval which consists in picking out the keyframes from the map only according to their geometric distance to the vehicle pose is not suitable for long-term operation. For example, if we perform a re-localization in the day while the closest keyframe to our current vehicle pose was taken during the night, the matching is extremely difficult; therefore, we have proposed to implement a ranking function which is able to consider other important criteria such as environmental conditions for the retrieval of keyframes.

We have calculated the matrix F_c in the beginning, the goal in the rest of the trajectory is to find the keyframe K^* that maximizes the probability of matching defined in Equation (3):

$$K^* = \underset{K}{\operatorname{argmax}}\, P(pt \in I_i, K) \tag{5}$$

Searching for K^* in the whole map is costly. From each traversal, we pick the keyframe $K^l, l \in [1, N]$ which has the minimum geometric distance to the current vehicle pose estimation. For all keyframes $\{K^l\}, l \in [1, N]$, we calculate the probability of matching with Equation (3) to retrieve the one with the highest score (K^*) with Equation (5). K^* will be used for further matching and pose calculating and optimizing.

In Section 6.1.2, we are comparing results obtained by the classic ranking function $\tau_1 = f_{dist}(I_i, K_j)$ (The score will be assigned based on the geometric distance between the two images as demonstrated in Figure 5) with results obtained by our proposed ranking function $\tau_2 = P(pt \in I_i, K)$ (Equation (3)).

3.2.3 Update of $F_c(trav(I_i), trav(K_j))$

It is interesting to update the values of the matrix F_c regularly after the first few meters of the trajectory. Indeed, this update can be useful when the state of the environment changes between the beginning and the end of the sequence. This update process works in parallel with the keyframes retrieval process. For this reason, we aim to avoid slowing down the localization by updating F_c for a single traversal at each iteration. Algorithm 2 describes the update process of the matrix F_c and the keyframe retrieval mechanism using our ranking function which has already been computed in the first few meters of the trajectory.

4 Map management

In this section we address the map update paradigm. We extended the mapping system of the aforementioned framework to adapt its maps to achieve long-term operations. The idea is to reduce the size of the map while keeping it as diverse as possible to cover the maximum number of different environmental conditions.

Algorithm 2: Keyframe retrieval and update of F_c

Parameters: The update rate $\alpha \leftarrow 0.1$
Steps:
$c \leftarrow N + 1$ // the index of the current traversal
$l \leftarrow 1$ // the index of the first traversal
for each current image I_i **do**
    Predict the pose $\hat{p}_i$ of I_i using the last computed pose p_{i-1} and the wheel odometry data.
    // Update of F_c:
    Pick from traversal l the closest keyframe to $\hat{p}_i$: K^l
    Match I_i to K^l and compute the inlier rate: $P(pt \in I_i, K^l)$
    Calculate x such as: $x \leftarrow \dfrac{P(pt \in I_i, K^l)}{f_{dist}(I_i, K^l)}$
    Update F_c: $F_c(c, l) \leftarrow (1 - \alpha)F_c(c, l) + \alpha x$
    $l \leftarrow l + 1$
    if $l > N$ **then**
        $l \leftarrow 1$
    end if
    // Keyframe retrieval and pose calculation:
    Retrieve from the map the keyframe K^* which has the highest score assigned by the ranking function:
$$K^* = \underset{K}{\operatorname{argmax}}\, P(pt \in I_i, K)$$
    Calculate the pose p_i by matching I_i and K^*
end for

Our proposed map management procedure consists in limiting the total number of traversals in the map (N) to a predefined number of traversals $\hat{N}$. When the number of traversals in the map N exceeds the predefined number $\hat{N}$ ($N = \hat{N} + 1$), our algorithm has to choose a traversal to remove from the map. The choice depends mainly on the matrix F_c defined in Section 3.2.1. In view of the fact that F_c is updating regularly along the traversal (as explained in Section 3.2.3), in the end of each traversal, we compute its average:

$$\bar{F_c} = \frac{1}{n}\sum_{i=1}^{n} F_c^i \tag{6}$$

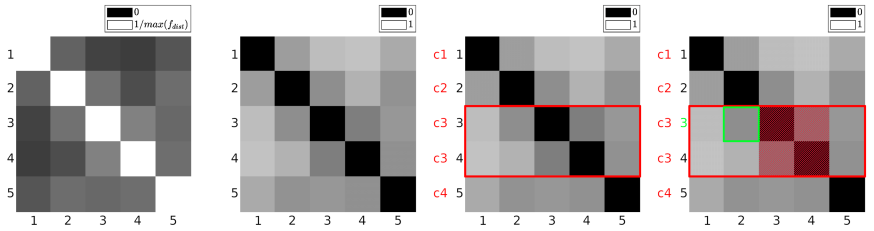
where n is the total number of images in the trajectory and F_c^i is the value of the matrix F_c at iteration i.

Our approach consists in using this matrix $\bar{F_c}$ to select the traversal which has the most resemblance to the others and then removing it from the map.

We used the hierarchical clustering algorithm to classify $\bar{F_c}$ in order to select the traversal to remove as explained in Figure 6.

Figure 6 shows the different steps of the traversal selection:

(a) We use the matrix $\bar{F_c}$ calculated with Equation (6) as the input of our hierarchical clustering algorithm.



(a) $\bar{F}_c$ (b) D (c) Clustering D (d) searching for min

Fig. 6: Steps for the selection of the traversal to remove.

(b) To perform hierarchical clustering on the matrix $\bar{F}_c$, we first need to convert it into a distance matrix. Since $\bar{F}_c$ is symmetric and its values are included between 0 and $1/max(f_{dist})$, we can get a distance matrix by normalizing it as follows:

$$D = J - \bar{F}_c \cdot max(f_{dist}) \tag{7}$$

J is a matrix of all ones with the same dimension as $\bar{F}_c$ ($[N \times N]$). The resulting matrix D is a distance matrix with zero diagonal and its values belong to $[0, 1]$.

(c) We classify D into $N-1$ classes using the hierarchical clustering algorithm ($N = 5$ in this example), this will result in classifying the two traversals i and j having the highest similarity in the same class ($i = 3$ and $j = 4$ in this example).

(d) In this step, we want to remove a traversal from the map while maintaining it as diverse as possible. To do so, we have to remove either the traversal i or the traversal j. To choose which one to remove, we search which one of them has more similarity to the other traversals. This is why we search for the minimum value in the i^{th} and j^{th} rows of the matrix while ignoring the diagonal and the elements with coordinates (i, j) and (j, i) (the hatched area in the figure). After finding the minimum, we remove its corresponding traversal (traversal 3 in the example).

5 Datasets

Intensive work on SLAM algorithms has produced a large number of related datasets such as KiTTi [12], Cityscapes [9].... The vast majority of these datasets are designed for localization in static environments with very small environmental change. However, datasets with many environmental conditions are required for applications that aim for long-term localization in dynamic environments. Datasets like Oxford RobotCar [19], NCLT [6] and UTBM [33] are widely used datasets for long-term localization applications since they include different environmental conditions. Both of the last two mentioned datasets contain only a few number of different sequences, which makes it

difficult for us to test our approach on them. For this reason, we are using only Oxford RobotCar and our own dataset on the test phase.

In Oxford RobotCar dataset, the itinerary and the direction of traversal followed during individual recordings vary between the different sequences. Accordingly, we have identified 20 sequences passing on the same route of ∼1.6km length. All these sequences are day-time sequences with only one dusk-time and 2 night-time sequences. The dusk-time sequence was recorded at the very beginning of the sunset time (16:34) and it was not possible for us to use it to properly localize the night-time sequences in a map built by day-time sequences. Moreover, we are interested to test the effect of lateral and angular deviation between sequences on the localization performance. However, the Oxford RobotCar dataset does not provide sequences with such criteria. This is the main reason that led us to record our own dataset.

The IPLT dataset was created from recorded images of two gray-scale 100° FOV cameras mounted on our experimental vehicle (one front and one rear camera) and wheel-odometry.

For Oxford RobotCar dataset, we have used only day-time sequences (with varying weather conditions) due the lack of intermediate dusk sequences to match day-time and night-time sequences. We used the visual odometry together with both front left and right cameras in our mapping framework.

6 Experiments and results

6.1 Keyframes retrieval

6.1.1 Experiments

We divide each dataset into mapping sequences and test sequences. We first proceeded to a mapping phase in which we built a global map consisting of N traversals from the mapping sequences with varying environmental conditions. We used the generated global maps as reference maps in our experiments in order to perform re-localization with the test sequences. This allowed us to evaluate the effectiveness of our algorithm in different conditions. We are also comparing results obtained while using $\tau_1 = f_{dist}(I_i, K_j)$ (which takes into account only the geometric distance as a criterion to assign scores to keyframes) with results obtained while using our proposed ranking function $\tau_2 = P(pt \in I_i, K_j) = f_{dist}(I_i, K_j) \cdot F_c(trav(I_i), trav(K_j))$ (which takes into consideration the geometric distance and the environmental conditions of the keyframes).

We present the average number of inliers observed in each sequence as well as the number of localization failures as criteria for the comparison. Practically, we found that the localization can be counted as reliable when there are at least 30 points matched between the current image and the database, below this threshold, we consider a localization failure. This is a conservative threshold to ensure the security of our autonomous shuttle [29].

The number of meters m required to initialize the matrix F_c (in the start of the trajectory) was chosen experimentally to minimise the distance between the value of F_c calculated after m meters and its mean $\bar{F}_c$ along the trajectory. In Table 2, we illustrate an example of this distance ($|F_c - \bar{F}_c|$) for different value of m.

Table 2: Value of $|F_c - \bar{F}_c|$ with respect to the choice of m.

m (meters)	5	10	20	30
$\lvert F_c - \bar{F}_c \rvert$	0.985	0.066	0.027	0.024

According to Table 2, we have chosen to fix the parameter m to 20 meters. Thus, we devote the first 20 meters of the trajectory for the calculation of the matrix F_c.

IPLT dataset From this dataset, we have selected 103 sequences with varying day-times, weather conditions (rain, snow, fog...) and some lateral and angular deviations, 10 of them were used for construction of the first global map (see Figure 7).

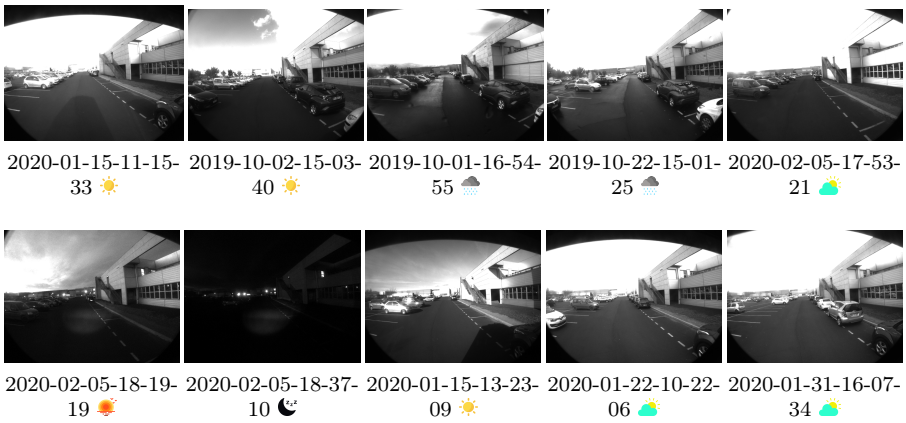


Fig. 7: An overview of images from IPLT dataset used for the construction of the first global map. For each sequence we are indicating the acquisition date and symbolizing the environmental condition by a small icon. Please refer to Table 3 for more details about the designation of condition icons used in this paper.

It is difficult to perform re-localization using night-time sequences on a map containing only day-time sequences, hence, we added a dusk sequence to the map to work as an intermediate between day and night sequences. This

Table 3: Designation of condition icons.

Condition	Sunny	Dusk	Night	Cloudy	Rainy	Foggy	Snowy
Icon							

allowed the SLAM algorithm to find matches linking all the traversals so all the poses have been optimized in the same bundle adjustment. We verified manually the map to make sure that all the poses are geometrically coherent. This means that there is no differential drift between the traversals. This global map contains sequences having different environmental conditions, it also includes some lateral and angular deviations between the sequences as illustrated in Figure 2.

Oxford RobotCar dataset We picked 8 sequences from this dataset to build our second global map (the Oxford map) while we used 12 other sequences for our tests. Figure 8 illustrates an overview of images taken from the sequences used for the construction of the Oxford map.

Fig. 8: An overview of images from Oxford RobotCar dataset used for the construction of the Oxford map.

6.1.2 Results

In this section, we demonstrate the efficiency of our keyframes retrieval approach by analyzing results obtained after performing re-localization with different sequences. As we mentioned in the previous section, we performed our tests on the two global maps obtained from the two datasets.

IPLT dataset We are using 93 sequences having different environmental conditions from IPLT dataset for this test phase. In Figure 9 we present an illustration of images extracted from some of these sequences.

Fig. 9: An overview of images recorded with the front camera for some of the sequences of IPLT dataset used in our tests.

In Figure 10, we present a comparison between localization performance on the IPLT map while using τ_1 and τ_2 as ranking functions. This figure

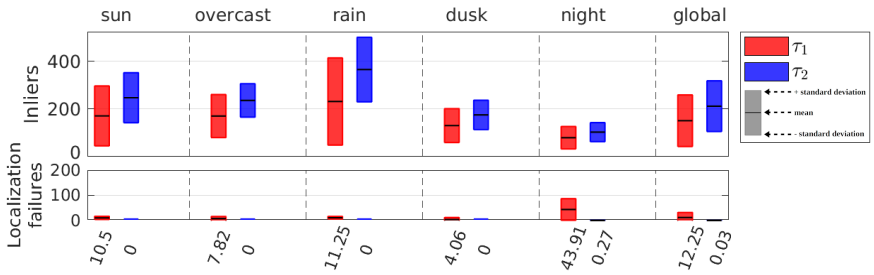


Fig. 10: A comparison between the two ranking functions τ_1 and τ_2 on the IPLT map using the 93 test sequences from IPLT dataset. The comparison is done according to two criterion: the average number of inliers per image and the average number of localization failures per sequence. Each box represents the mean value + and - the standard deviation of inliers (or localization failures) recorded while performing re-localization using all the sequences of the corresponding class on the global map. The color of the boxes indicates which ranking function was used to record these values. For better readability of localization failures values, we plot the average number of localization failures in the bottom x-axis.

was generated using all the 93 test sequences. These sequences were classified manually into 5 different classes according to their environmental condition. These 5 classes (sun, overcast, rain, dusk and night) are containing respectively 13, 39, 12, 17 and 11 sequences. The "global" class regroups all the 93 sequences. For each class, we show the average number of inliers observed on the sequences of this class. We also present the average number of localization failures experienced on these sequences.

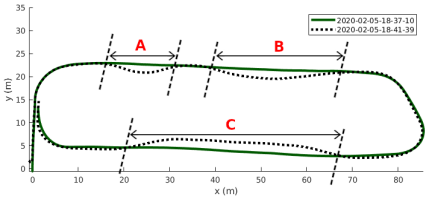
Overall, we note that our proposed keyframes retrieval approach has significantly increased the number of inliers observed during re-localization for all classes. We also notice that with our approach, we have successfully reduced the number of bad localizations per sequence from 12.25 to only 0.03.

In Figure 11, we demonstrate that our ranking function τ_2 depends on both environmental conditions and geometric distance to assign scores to keyframes which helps to retrieve good keyframes for localization. We performed a re-localization on the IPLT map with the test sequence 2020-02-05-18-41-39 . The global map contains a sequence recorded a few minutes before this test sequence (2020-02-05-18-37-10) and both of them have similar environment conditions.

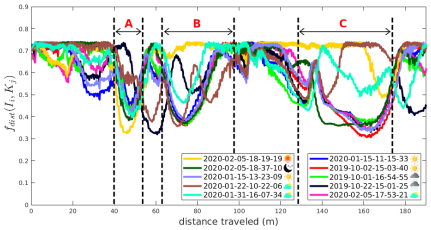
In segments A, B and C, the two sequences have some lateral and angular deviations (Figure 11a), thus, τ_1 was not able to retrieve keyframes from traversals with similar environmental condition in those zones (Figure 11b), while our proposed ranking function τ_2 has successfully retrieved good keyframes even with the deviation (Figure 11c).

In Section 3.2.3, we presented the update paradigm of the matrix F_c, however, the sequences used in the previous tests are quite short and they don't contain significant changes in the environmental condition and consequently they can't prove the interest of this update. For this reason, we recorded a long sequence (2019-12-05-16-43-56) which has multiple loops in the parking lot starting from 16:44 until 17:54 and it includes day, dusk and night conditions. In Figure 12, we inspect the update process of the matrix F_c. We plot the values located in the last row (or column) of the matrix F_c along a localization session using this long sequence. The last row (with index $N+1$) of this matrix is referring to the current traversal (the sequence 2019-12-05-16-43-56). Therefore, this row contains the similarity scores between the current images (images of the current traversal) and the closest keyframe from each one of the N traversals in the map.

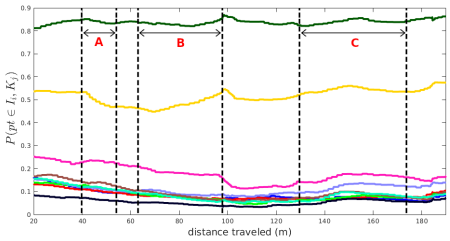
We observe that in the start of the trajectory (before the dusk), the curve representing the traversal 2020-02-05-17-53-21 (the pink curve) is at the top. This means that this traversal has the highest similarity with the current environmental conditions. Therefore, keyframes from this traversal will be used for localization (we want to point that $P(pt \in I_i, K_j)$ is also taking into account the geometrical distance. Even though the advantage of this traversal according to F_c, our ranking function can still prefer retrieving keyframes from other traversals if f_{dist} is assigned low scores for keyframes of this traversal). In the dusk time, we notice a decrease in the values of the pink curve and an increase in the yellow one (2020-02-05-18-19-19). This means that F_c is



(a) Path traveled for the sequence 2020-02-05-18-37-10 and the sequence 2020-02-05-18-41-39.



(b) Values assigned to keyframes by τ_1 while re-localizing with the sequence (i).



(c) Values assigned to keyframes by τ_2 while re-localizing with the sequence (i).

Fig. 11: Effect of lateral and angular deviations on the two ranking functions τ_1 and τ_2. In sub-figure (a), the green path represents the sequence 2020-02-05-18-37-10 from the map, while the dashed path represents the sequence (i) from the test sequences. The two sequences have similar environmental conditions but with some deviations between them. The choice of the ranking function is illustrated in sub-figure (b) and (c). Each traversal in the map is represented by a curve with a different color. For each traversal, the curve indicates the computed probability of matching the current image with the nearest keyframe of this traversal. The colors of the curves in sub-figure (c) are the same as in sub-figure (b).

indicating that there are more resemblance with the traversal represented by the yellow curve (which was recorded in the dusk). Finally, at the beginning of the night-time, we consider a notable decrease in all the curves except for the dark green one (2020-02-02-18-37-10 $\mathbf{C}_{z^z}$). This is the only traversal recorded in the night, and accordingly, F_c is recommending retrieving keyframes from this traversal.

Oxford RobotCar dataset To evaluate the performance of our approach on the Oxford RobotCar dataset, we used 12 test sequences with varying environmental conditions. These 12 sequences were classified into 4 classes: sun, overcast, rain and snow. The sun class contains 5 sequences, the overcast contains 4 sequences, the rain contains 2 sequences and the snow class contains only 1 sequence. In Figure 13, we are comparing the re-localization performance of τ_1 and τ_2 with respect to the average number of inliers and localization failures as we did for IPLT dataset.

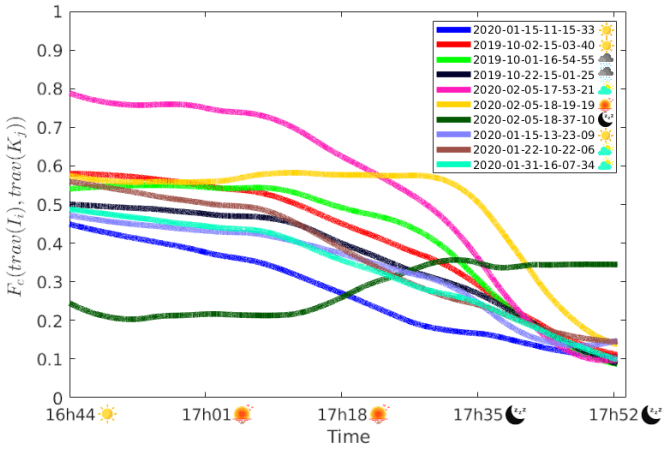


Fig. 12: The values of F_c along a one hour and 10 minutes sequence containing day, dusk and night conditions. Each curve corresponds to a traversal in the map (N curves for N traversals), and the values in each curve are indicating the similarity scores between the images of the current traversal and their closest keyframes that belong to the traversal represented by this curve. These curves were smoothed for better readability.

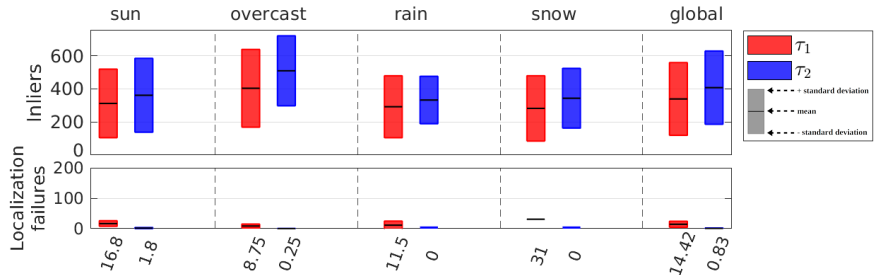


Fig. 13: A comparison between the two ranking functions τ_1 and τ_2 on the Oxford global map using the 12 test sequences from Oxford RobotCar dataset.

According to the figure, τ_2 has managed to increase the number of inliers and decrease the number of localization failures from more than 14 failures per sequence to less than 1. In this dataset, we consider a less significant increase in term of inliers compared to the increase observed on IPLT dataset (Figure 10). This is due to the fact that the IPLT dataset contains more sequences with more changes in environmental conditions and it incorporates some sequences with lateral and angular deviations that seriously impact the performance of τ_1.

We also note that this dataset contains a significant number of overexposed images especially in sunny sequences [14] (thus the 9 localization failures for τ_2 in the sun class). We present some examples of these images in Figure 14.

2014-11-18-13-20-12 ☀ | 2015-02-06-13-57-16 ☀ | 2015-05-29-09-36-29 🌧 | 2015-07-29-13-09-26 ☀

Fig. 14: Example of extremely overexposed images that led to localization errors.

6.2 Map management

6.2.1 Experiments

We have tested our map management approach on the same datasets used in the previous section. We also have used the same division for the mapping sequences and test sequences.

As we mentioned in the Section 4, we aim to limit the number of traversals used to build the global map to a predefined number $\hat{N}$. We call an $\hat{N}$-session map a map constituted from $\hat{N}$ traversals. In other words, our approach has to choose the best $\hat{N}$-session map from a total of N traversals ($N = 10$ for the Oxford global map and $N = 8$ for the IPLT global map).

To evaluate the efficiency of our approach, we first proceed to build all possible maps constituted of $\hat{N}$ traversals ($\hat{N} = 3$ or $\hat{N} = 4$ in our tests). This will result in obtaining n different $\hat{N}$-session maps:

$$n = {}_{N}C_{\hat{N}} = \frac{N!}{(N-\hat{N})!\,\hat{N}!} \tag{8}$$

The goal is to determine if there exists a small map that can be used to localize every test sequence without significantly increasing the failure rate of the localization compared to the map which contains all the traversals. It will also allow to rank all the possible $\hat{N}$-session maps, which means that we can position the result of our approach among all the n possibilities. Thus we'll be able to compare the result of our incremental approach with the global optimal $\hat{N}$-session map. Of course, this is possible only for an evaluation purpose, the exhaustive search of the global optimum is intractable for real use cases with hundreds of traversals.

In order to compute localization error, the n generated $\hat{N}$-session maps are evaluated by performing re-localization with the test sequences. Since we do not possess the ground-truth poses to compute the error in the IPLT dataset, for each test sequence, we create the corresponding ground-truth poses by performing re-localization on the global map while retrieving from the map the 4 keyframes which have the highest probability of matching $P(pt \in I_i, K_j)$. The landmarks of these 4 retrieved keyframes are matched to the current image to calculate the ground-truth pose using PnP+RANSAC. The Oxford RobotCar dataset is providing the RTK ground-truth poses [20]. However, we

deplore a lack of ground-truth poses for some of the sequences used in this paper (either a total or a partial lack, e.g. 2014-11-18-13-20-12, 2014-12-16-09-14-09, 015-02-03-08-45-10, etc.). Also, the ground-truth poses provided by this dataset comprehend a large error margin (∼15 cm in latitude and longitude) which is an order of magnitude higher than the local accuracy usually given by visual localization in such applications. Therefore, we decided to proceed to the same way of ground-truth calculation as we did for the IPLT dataset.

To compute the localization error, we perform re-localization with the test sequences on all the n $\hat{N}$-session maps. For each $\hat{N}$-session map, we compute the average of localization errors of all the test sequences. The localization errors correspond to the euclidean distance between the ground-truth poses calculated on the global map and the poses computed while re-localizing on the $\hat{N}$-session maps.

Afterwards, we proceed to an incremental search step. In this step, we employ our map management approach to build a map from $\hat{N}$ traversals among N traversals. Considering the fact that the order in which the traversals are added to the map can influence the resulting map, we test our approach with $100,000$ different orders of the N traversals.

6.2.2 Results

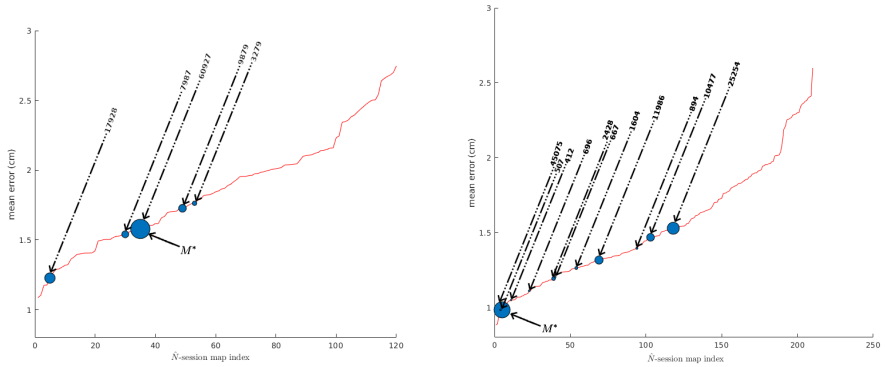
In this section, we show the efficiency of our map management approach. As in the keyframes retrieval approach, we test our map management with two datasets.

IPLT dataset We evaluated our approach on the IPLT dataset using the global map presented in Figure 7 using two different values of $\hat{N}$ to demonstrate its efficiency on different setups: $\hat{N} = 3$ and $\hat{N} = 4$.

In Figure 15, we present the result of our map management approach which consists of the average of localization error of all the n $\hat{N}$-session maps composed from $\hat{N}$ traversals. We also point to the $\hat{N}$-session maps that are selected by our approach.

This figure shows that the localization error has decreased globally when we passed from $\hat{N} = 3$ to $\hat{N} = 4$ (which is clear by the red curve in the two sub-figures). In the first sub-figure (15a), $\hat{N}$ is equal to 3. According to Equation (8), the number of all possible $\hat{N}$-session maps composed from 3 traversals is $n = 120$. In the second sub-figure (15b), $\hat{N}$ is equal to 4, accordingly $n = 210$. As visible in both sub-figures, our approach has produced multiple results as consequence of using 100,000 different permutations of the N traversals as input.

In order to evaluate the influence of the map compression in the localization performance, we compare the performance of localization on the initial global map M_0 which is composed of N traversals with performance of localization on the most reproduced $\hat{N}$-session map M^* (the biggest dot in each sub-figure). In Figure 16 we present the average number of inliers per image and the average number of localization failures per sequence recorded while re-localizing on



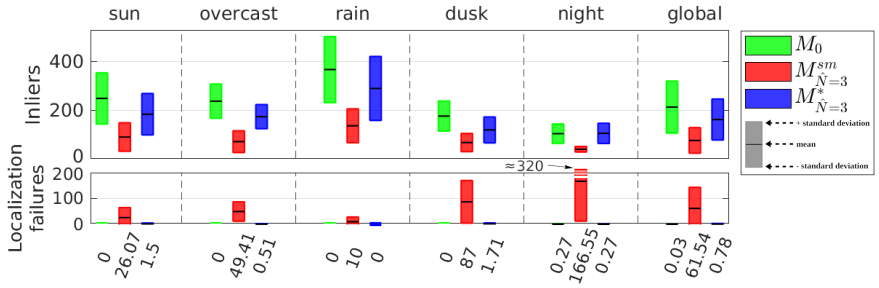
(a) $\hat{N} = 3$ (b) $\hat{N} = 4$

Fig. 15: Average localization errors for the IPLT dataset global map. The red curve designates the average of localization errors obtained while performing re-localization with the test sequences on the n $\hat{N}$-session maps. All the $\hat{N}$-session maps are sorted in ascending order according to their corresponding errors. The blue dots are pointing to the results of our map management approach on 100,000 different permutations of the N traversals. We specify the number of times each result has been reproduced by our approach with the size of its corresponding dot and with the annotations on the figures.

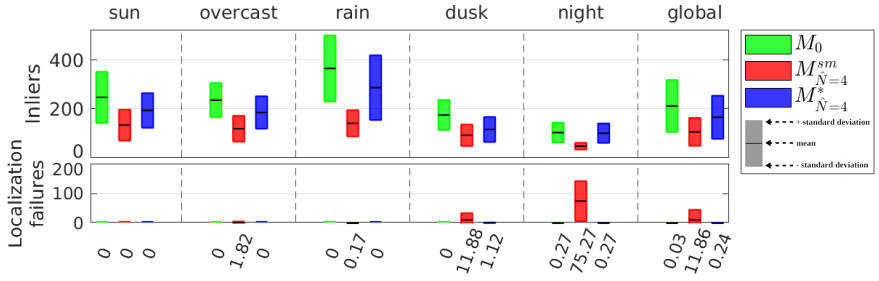
the initial map M_0 and on both $\hat{N}$-session maps ($M^*_{\hat{N}=3}$ and $M^*_{\hat{N}=4}$). We also present a comparison with the state-of-the-art "Summary Maps" approach proposed by Muhlfellner *et al.* [22]. In their approach, Muhlfellner *et al.* are scoring landmarks according to the number of different localization sessions in which they appear and they are removing the landmarks with the lowest scores in an offline process. To guarantee a fair comparison with their solution, we remove the landmarks having the lowest scores from M_0 until we get a map M^{sm} having the desired size. This means $M^*_{\hat{N}=3}$ and $M^{sm}_{\hat{N}=3}$ have the same number of landmarks and thus both maps have approximately the same size. The same thing is valid for $M^*_{\hat{N}=4}$ and $M^{sm}_{\hat{N}=4}$.

It is clear from the figure that the localization performance has slightly degraded after limiting the map to $\hat{N} = 4$ traversals and has degraded more for $\hat{N} = 3$ traversals. However, even with this degradation, the localization can still be considered as reliable especially for $\hat{N} = 4$ as the number of localization failures (0.24 localization failures per sequence, i.e. 22 failures in total, for $M^*_{\hat{N}=4}$) remains insignificant. This means that after compressing the map, only 22 images among all the 93 sequences (which incorporate a total of 159,074 images) where matched with less than 30 inliers. The map $M^*_{\hat{N}=4}$ does not include any dusk sequences, thus, we consider a slight degradation in term of localization failures for the dusk sequences (from 0 failures per sequence for M_0 to 1.12 for $M^*_{\hat{N}=4}$).

In the other hand, the map M^{sm} has shown a major weakness in localization especially with night sequences. This can be explained by the fact that our first global map M_0 contains only one night sequence (Figure 7) which



(a) $\hat{N} = 3$



(b) $\hat{N} = 4$

Fig. 16: Localization performance comparison on M_0, M^* and on the map M^{sm} generated with the Summary Maps approach [22] using the 93 test sequences from IPLT dataset. Subfigures (a), (b) represent the localization performance when choosing 3 and 4 respectively as values for the parameter $\hat{N}$.

means that landmarks observed in the night will have a low score since they have appeared only in one session.

In Table 4, we plot the memory occupancy and the number of landmarks included in each map used to evaluate our approach on the IPLT dataset.

Table 4: Memory occupancy and number of landmarks in each map built using IPLT dataset.

map	M_0	$M^*_{\hat{N}=4}$	$M^{sm}_{\hat{N}=4}$	$M^*_{\hat{N}=3}$	$M^{sm}_{\hat{N}=3}$
Memory occupancy (MB)	~ 450	~ 165	~ 185	~ 120	~ 135
Number of landmarks ($\times 10^6$)	~ 1.25	~ 0.5	~ 0.5	~ 0.36	~ 0.36

According to this table, $M^*_{\hat{N}=4}$ was obtained after compressing the map M_0 with more than 0.5 compressing rate. Despite this compression, we note that $M^*_{\hat{N}=4}$ was able to achieve a very competitive level of performance with the uncompressed map M_0. We also note that our approach has remarkably

outperformed a map with a similar compression rate that was generated with another approach ($M^{sm}_{\hat{N}=4}$).

Oxford RobotCar dataset As in the previous section, we evaluate our approach on the Oxford RobotCar dataset. The global map used here is the same as in Section 6.1.1 (Figure 8). In Figure 17, we present the average of localization error of all the n $\hat{N}$-session maps composed of $\hat{N} = 3$ traversals (according to Equation (8), $n = 56$). Since this global map is built only by $N = 8$ traversals, we exhibit the result of our approach on the all possible permutations of 8 traversals ($8! = 40320$ permutations).

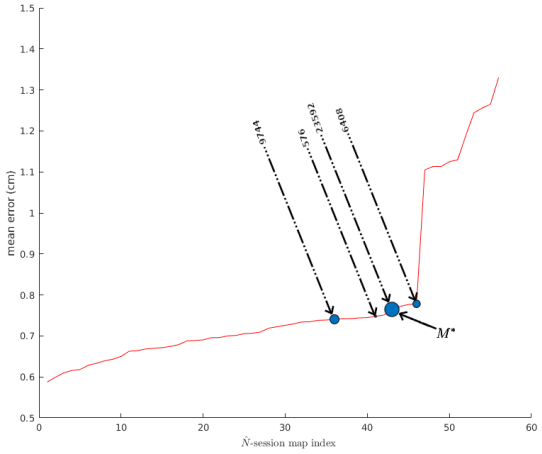


Fig. 17: Average localization errors of the the Oxford RobotCar dataset global map with $\hat{N} = 3$.

In Figure 18 we present a comparison between M_0, $M^*_{\hat{N}=3}$ and $M^{sm}_{\hat{N}=3}$ in term of inliers average and localization failures average. This figure shows

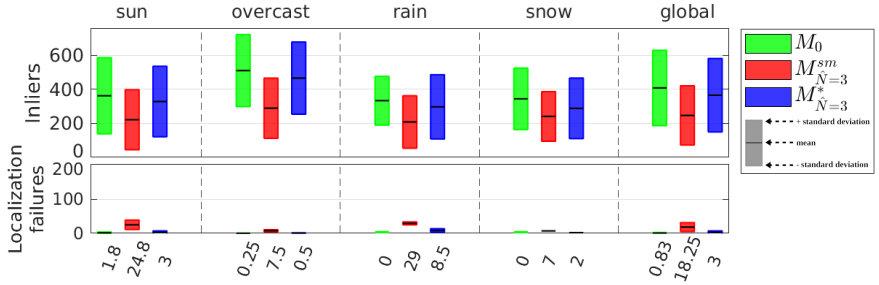


Fig. 18: Localization performance comparison on M_0, $M^*_{\hat{N}=3}$ and on $M^{sm}_{\hat{N}=3}$ using the 12 test sequences from Oxford RobotCar dataset.

that the performance of localization on Oxford RobotCar dataset has slightly degraded when using our approach with $\hat{N} = 3$. This kind of degradation is expected after compressing the map from 8 traversals to only 3. According to the figure, there is a remarkable gap between localization performance on $M^{*}_{\hat{N}=3}$ and on $M^{sm}_{\hat{N}=3}$.

In Table 5, we present the memory occupancy and the number of landmarks included in each map used to evaluate our approach on the Oxford RobotCar dataset.

Table 5: Memory occupancy and number of landmarks in each map built using Oxford RobotCar dataset.

map	M_0	$M^{*}_{\hat{N}=3}$	$M^{sm}_{\hat{N}=3}$
Memory occupancy (MB)	~ 970	~ 330	~ 430
Number of landmarks ($\times 10^6$)	~ 14	~ 5	~ 5

7 Conclusion

In this paper we have presented an algorithm which can be used for long-term localization process. Our proposed algorithm is composed of two complementary processes. The first process is used to retrieve keyframes based on their euclidean distance to the current image and the environmental condition. This keyframes retrieval consists of using probabilistic ranking function that exploits information collected during the first few meters of the trajectory to determine whether or not a keyframe is suitable for localization. Our second process aims to bound the map's size to avoid its continued inflation. Our experiments demonstrated, on two different datasets, that our ranking function was able to retrieve good keyframes in different environmental conditions which in turn helped to improve localization performance. We have also shown that the localization performance has not decreased significantly after reducing the size of the map with our proposed map management approach. Finally, we have presented a new dataset that contains challenging environmental conditions which we make available to the community in the hope that it will be useful to other people working in this field.

Acknowledgements This work has been sponsored by the French government research program "Investissements d'Avenir" through the IMobS3 Laboratory of Excellence (ANR-10-LABX-16-01) and the RobotEx Equipment of Excellence (ANR-10-EQPX-44), by the European Union through the Regional Competitiveness and Employment program 2014-2020 (ERDF - AURA region) and by the AURA region.

Conflict of interest

We declare that authors have no conflict of interest on this work.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European conference on computer vision. pp. 404–417. Springer (2006)
2. Bouaziz, Y., Royer, E., Bresson, G., Dhome, M.: Keyframes retrieval for robust long-term visual localization in changing conditions. In: to appear in 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI). IEEE (2021)
3. Bürki, M., Cadena, C., Gilitschenski, I., Siegwart, R., Nieto, J.: Appearance-based landmark selection for visual localization. Journal of Field Robotics **36**(6), 1041–1073 (2019)
4. Bürki, M., Dymczyk, M., Gilitschenski, I., Cadena, C., Siegwart, R., Nieto, J.: Map management for efficient long-term visual localization in outdoor environments. In: 2018 IEEE Intelligent Vehicles Symposium (IV). pp. 682–688. IEEE (2018)
5. Bürki, M., Gilitschenski, I., Stumm, E., Siegwart, R., Nieto, J.: Appearance-based landmark selection for efficient long-term visual localization. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4137–4143. IEEE (2016)
6. Carlevaris-Bianco, N., Ushani, A.K., Eustice, R.M.: University of Michigan North Campus long-term vision and lidar dataset. International Journal of Robotics Research **35**(9), 1023–1035 (2015)
7. Churchill, W., Newman, P.: Practice makes perfect? managing and leveraging visual experiences for lifelong navigation. In: 2012 IEEE International Conference on Robotics and Automation. pp. 4525–4532. IEEE (2012)
8. Churchill, W., Newman, P.: Experience-based navigation for long-term localisation. The International Journal of Robotics Research **32**(14), 1645–1661 (2013)
9. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
10. Diaz-Escobar, J., Kober, V., Gonzalez-Fraga, J.A.: Luift: Luminance invariant feature transform. Mathematical Problems in Engineering **2018**, 1–17 (2018)
11. Dymczyk, M., Lynen, S., Cieslewski, T., Bosse, M., Siegwart, R., Furgale, P.: The gist of maps-summarizing experience for lifelong localization. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 2767–2773. IEEE (2015)
12. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
13. Harris, C.G., Stephens, M., et al.: A combined corner and edge detector. In: Alvey vision conference. vol. 15, pp. 10–5244. Citeseer (1988)
14. Jatzkowski, I., Wilke, D., Maurer, M.: A deep-learning approach for the detection of overexposure in automotive camera images. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). pp. 2030–2035. IEEE (2018)
15. Lébraly, P., Royer, E., Ait-Aider, O., Deymier, C., Dhome, M.: Fast calibration of embedded non-overlapping cameras. In: 2011 IEEE international conference on robotics and automation. pp. 221–227. IEEE (2011)
16. Linegar, C., Churchill, W., Newman, P.: Work smart, not hard: Recalling relevant experiences for vast-scale but time-constrained localisation. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 90–97. IEEE (2015)
17. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
18. MacTavish, K., Paton, M., Barfoot, T.D.: Selective memory: Recalling relevant experience for long-term visual localization. Journal of Field Robotics **35**(8), 1265–1292 (2018)
19. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 year, 1000km: The oxford robotcar dataset. The International Journal of Robotics Research (IJRR) **36**(1), 3–15 (2017). https://doi.org/10.1177/0278364916679498, `http://dx.doi.org/10.1177/0278364916679498`
20. Maddern, W., Pascoe, G., Gadd, M., Barnes, D., Yeomans, B., Newman, P.: Real-time kinematic ground truth for the oxford robotcar dataset. arXiv preprint arXiv: 2002.10152 (2020), `https://arxiv.org/pdf/2002.10152`

21. Milford, M.J., Wyeth, G.F.: Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In: 2012 IEEE International Conference on Robotics and Automation. pp. 1643–1649. IEEE (2012)
22. Mühlfellner, P., Bürki, M., Bosse, M., Derendarz, W., Philippsen, R., Furgale, P.: Summary maps for lifelong visual localization. Journal of Field Robotics **33**(5), 561–590 (2016)
23. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)
24. Murillo, A.C., Kosecka, J.: Experiments in place recognition using gist panoramas. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops. pp. 2196–2203. IEEE (2009)
25. Naseer, T., Oliveira, G.L., Brox, T., Burgard, W.: Semantics-aware visual localization under challenging perceptual conditions. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 2614–2620. IEEE (2017)
26. Pascoe, G., Maddern, W., Tanner, M., Piniés, P., Newman, P.: Nid-slam: Robust monocular slam using normalised information distance. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1435–1444 (2017)
27. Pepperell, E., Corke, P., Milford, M.: Routed roads: Probabilistic vision-based place recognition for changing conditions, split streets and varied viewpoints. The International Journal of Robotics Research **35**(9), 1057–1179 (2016)
28. Piasco, N., Sidibé, D., Gouet-Brunet, V., Demonceaux, C.: Learning scene geometry for visual localization in challenging conditions. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 9094–9100 (May 2019). https://doi.org/10.1109/ICRA.2019.8794221
29. Royer, E., Marmoiton, F., Alizon, S., Ramadasan, D., Slade, M., Nizard, A., Dhome, M., Thuilot, B., Bonjean, F.: Lessons learned after more than 1000 km in an autonomous shuttle guided by vision. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). pp. 2248–2253. IEEE (2016)
30. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: Robust hierarchical localization at large scale. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12716–12725 (2019)
31. Schneider, T., Dymczyk, M., Fehr, M., Egger, K., Lynen, S., Gilitschenski, I., Siegwart, R.: maplab: An open framework for research in visual-inertial mapping and localization. IEEE Robotics and Automation Letters **3**(3), 1418–1425 (2018)
32. Tian, Y., Yu, X., Fan, B., Wu, F., Heijnen, H., Balntas, V.: Sosnet: Second order similarity regularization for local descriptor learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11016–11025 (2019)
33. Yan, Z., Sun, L., Krajnik, T., Ruichek, Y.: Eu long-term dataset with multiple sensors for autonomous driving. arXiv preprint arXiv:1909.03330 (2019)
34. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: Learned invariant feature transform. In: European Conference on Computer Vision. pp. 467–483. Springer (2016)